Advances in Knowledge-Based Technologies

Proceedings of the
Master and PhD Seminar
Winter term 2014/15, part 1

Softwarepark Hagenberg
SCCH, Room 0/2
24 November 2014

Software Competence Center Hagenberg
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 431
www.flll.jku.at

Program

Session 1. Chair: Susanne Saminger-Platz

8:30	Peter Smrek:
	A note on several kinds of the level dependent capacities-based Sugeno integrals
9:00	Ladislav Šipeky:
	On some transformations of aggregation functions
9:30	Tomáš Bacigál:
	Archimedean copulas, additive generators and distance functions
10:00	Michal Dibala:
	Conic and Archimax copulas

Session 2. Chair: Bernhard Moser

10:45	Edwin Lughofer:
	Dynamic Inclusion of New Event Types in Visual Inspection using Evolving Fuzzy Classifiers
11:15	Francisco Serdio:
	Expanding the set of Residual Generators by Genetic-Fuzzy Systems
11:45	Mariya Zhariy:
	Efficient Sensor Placement in Flow Networks and Sensor Networks
12:15	Thomas Grubinger:
	Domain Generalization based on Transfer Component Analysis

A note on several kinds of the level dependent capacities-based Sugeno integrals

SMREK Peter[1] and STUPŇANOVÁ Andrea[2]

[1] *Slovak University of Technology in Bratislava*
Faculty of Civil Engineering, Radlinského 11, 81368 Bratislava
Slovakia
peter.smrek@stuba.sk

[2] *Faculty of Civil Engineering, Radlinského 11, 81368 Bratislava*
Slovakia
andrea.stupnanova@stuba.sk

One can find several equivalent definitions of the standard Sugeno integral, based on clasical capacities (i.e. monotone set function satisfying the boundary condition $m(\emptyset) = 0$ and $m(X) = 1$), see [6], for equivalent definitions see also [1] and [5].
In this contribution we present possible generalized forms of Sugeno integrals based on level dependent capacities, which are not equivalent in general, and their relationships.

Consider an arbitrary fix measurable space $(X, \mathcal{A})$ and denote by $\mathcal{F}$ the class of all $\mathcal{A}-$measurable functions $f : X \to [0, 1]$. If m is a capacity, then Sugeno integral is a functional $Su_m(f) : \mathcal{F} \to [0, 1]$, given by

$$Su_m(f) = \sup\{\min(a, m(A))|a \cdot 1_A \leq f\}. \tag{1}$$

Equivalently, Su_m can be expressed as

$$Su_m(f) = \sup\{\min(a, m(f \geq a))|a \in [0, 1]\}, \tag{2}$$

or

$$Su_m(f) = \sup\{\min(m(A), \min(f(x)|x \in A)|A \in \mathcal{A})\}. \tag{3}$$

In [3] and in [1] another equivalent definition of Sugeno integral was introduced, namely

$$Su_m(f) = \inf\{\max(a, m(f \geq a))|a \in [0, 1]\}. \tag{4}$$

The concept of capacities was extended to level dependent capacities [2], see also [4, 5].

Definition *Let* $(X, \mathcal{A})$ *be a measurable space. A mapping* $M : \mathcal{A} \times [0,1] \to [0,1]$ *such that for each* $t \in [0,1], M(\cdot, t) = m_t$ *is a capacity, is called a level dependent capacity.*

Klement at al. have proposed in [5] (see also [4]) the function $h_{m,f} : [0,1] \to [0,1]$

$$h_{m,f}(t) = m(f \geq t) = m(\{x \in X | f(x) \geq t\}),$$

which summarizes all information contained in a capacity m and in a measurable function f. The function $h_{M,f} : [0,1] \to [0,1]$ is given analogically by

$$h_{M,f}(t) = M(\{f \geq t\}, t) = m_t(f \geq t).$$

Then we can get the greatest Sugeno integral based on level dependent capacities $(Su_M)^* : \mathcal{F} \to [0,1]$, and the smallest one $(Su_M)_* : \mathcal{F} \to [0,1]$, respectively, given by

$$(Su_M)^*(f) = \sup\{\min(t, h_{M,f}(v)) | 0 \leq t \leq v \leq 1\} \tag{5}$$

and

$$(Su_M)_*(f) = \sup\{\min(t, h_{M,f}(u)) | 0 \leq u \leq t \leq 1\}. \tag{6}$$

If we generalize formulae (1)-(4) for level dependent capacities we get the next possible forms of level dependent capacities based Sugeno integral:

$$Su_M^{(1)}(f) = \sup\{\min(a, m_a(A)) | a \cdot 1_A \leq f\}, \tag{7}$$

$$Su_M^{(2)}(f) = \sup\{\min(a, m_a(f \geq a)) | a \in [0,1]\}, \tag{8}$$

$$Su_M^{(3)}(f) = \sup\{\min(t, m_t(A)) | A \in \mathcal{A}, t = \min(f(x) | x \in A)\}, \tag{9}$$

$$Su_M^{(4)}(f) = \inf\{\max(a, m_a(f \geq a)) | a \in [0,1]\}. \tag{10}$$

We can show that it holds

$$Su_M^{(1)}(f) = Su_M^{(2)}(f) = Su_M^{(3)}(f). \tag{11}$$

Moreover, we can consider the level dependent capacity-based universal Sugeno integral K^{Min} constructed similarly as minimum copula-based universal integral (see [5]). K^{Min} is given by

$$K^{\text{Min}}(M, f) = P_{\text{Min}}(\{(x, y) \in]0,1]^2 \mid y \leq h_{M,f}(x)\}). \tag{12}$$

Then we get

$$(Su_M)_*(f) \leq K^{\text{Min}}(M, f) \leq (Su_M)^*(f). \tag{13}$$

One can show on an example that both inequalities in (13) are strict.

Theorem *Let* $(X, \mathcal{A})$ *be a measurable space,* $f : X \to [0, 1]$ *a measurable function on* $(X, \mathcal{A})$*, and* M *a level dependent capacity. Then*

$$(Su_M)^*(f) = Su_M^{(i)}(f), i = 1, 2, 3 \quad \textit{and} \quad (Su_M)_*(f) = Su_M^{(4)}(f).$$

Summarizing, we have shown that there are exactly three different types of the level dependent capacities-based Sugeno integrals.

Acknowledgement
The support of the grant VEGA 1/0171/12 is kindly announced.

References

[1] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap: *Aggregation Functions,* Cambridge University Press, Cambridge, 2009.

[2] S. Greco, B. Matarazzo, S. Giove: *The Choquet integral with respect to a level dependent capacity,* Fuzzy Sets and Systems 175, 2011, pp. 1–35.

[3] A. Kandel, W.J. Byatt: *Fuzzy sets, fuzzy algebra, and fuzzy statistics,* Proceedings of the IEEE vol. 66 no.12, 1978, pp. 1619–1639.

[4] E.P. Klement, A. Kolesárová, R. Mesiar, A. Stupňanová: *A generalization of universal integrals by means of level dependent capacities,* Knowledge-Based Systems 38, 2013, pp. 14–18.

[5] E.P. Klement, R. Mesiar, E. Pap: *A universal integral as common frame for Choquet and Sugeno integral,* IEEE Transactions on Fuzzy Systems 18, 2010, pp. 178–187.

[6] M. Sugeno: *Theory of fuzzy integrals and its applications,* PhD thesis, Tokyo Institute of Technology, 1974.

[7] Z . Wang, G.J. Klir: *Generalized Measure Theory,* Springer, 2009.

On some transformations of aggregation functions

SIPEKY Ladislav

Department of Mathematics and Descriptive Geometry
Faculty of Civil Engineering
Slovak University of Technology
Radlinskeho 11
Bratislava, Slovakia
ladislav.sipeky@stuba.sk

We propose the concept of superadditive and of subadditive transformations of aggregation functions acting on non-negative reals, in particular of integrals with respect to monotone measures. We discuss the concept on various examples based on everyday needs connected to linear programming. Also the properties of the superadditive and of subadditive transformations are studied with connections to other integrals (for more details on integrals see [1], [2], [3]). Moreover, subadditive transformations of distinguished integrals are also discussed. In discrete cases we can speak about motivation from economics.

Acknowledgement
The support of the grant VEGA 1/0171/12 is kindly announced.

References

[1] Grabisch, M. and Marichal, J.L. and Mesiar, R. and Pap, E. *Aggregation Functions.* Cambridge University Press. (2009).

[2] Lehrer, E. *A new integral for capacities.* Economic Theory. **39** (2009) 157–176.

[3] Sugeno, M. *Theory of fuzzy integrals and its applications.* Ph.D. Thesis, Tokyo Institute of Technology. (1974).

Archimedean copulas, additive generators and distance functions

Tomáš Bacigál[1] and Mária Ždímalová[2]

Abstract

We discuss a new construction method for obtaining additive generators of Archimedean copulas of a fixed dimension n (or of any dimension $n \geq 2$) proposed by McNeil and Nešlehová [1], the so-called Williamson n-transform (and its limit Laplace transform). These methods are illustrated on several examples. We show that due to equivalence of the weak convergence of positive distance functions, of the pointwise convergence of related additive generators fixed in point 0.5, and of the pointwise convergence of related copulas, we may approximate any n-dimensional Archimedean copula by a transformation of a convex sum of Dirac distance functions concentrated in certain suitably chosen points.

Introduction

A function $C\colon [0,1]^n \to [0,1]$, $n \geq 2$ is called a (n-dimensional) copula whenever it satisfies boundary conditions (i.e. it has 0 as annihilator and 1 as neutral element) and it is an n-increasing function, for more details see, e.g., [2]. A copula C belongs to the class of Archimedean copulas whenever it can be generated by a n-monotone continuous strictly decreasing function $f\colon [0,1] \to [0,\infty]$, $f(1) = 0$, via

$$C(x_1, \ldots, x_n) = f^{(-1)}\left(\sum_{i=1}^{n} f(x_i)\right),$$

where $f^{(-1)}\colon [0,\infty] \to [0,1]$ given by $f^{(-1)}(u) = f^{-1}\big(\min(u, f(0))\big)$ is the pseudo-inverse.

McNeil and Nešlehová in [1] not only provides necessary and sufficient conditions for f to be an additive generator of an n-dimensional copula, they also describe in details an interesting link between additive generators f of Archimedean copulas C and positive distance functions F (distribution functions with support on $]0,\infty[$), which will be called the Williamson n-transform and is given by

$$F(x) = 1 - \sum_{k=0}^{n-2} \frac{x^k g^{(k)}(-x)}{k!} - \frac{x^{n-1} g_-^{(n-1)}(-x)}{(n-1)!}$$

[12]Faculty of Civil Engineering, Slovak university of Technology in Bratislava, Slovakia
Email: bacigal@math.sk, zdimalova@math.sk

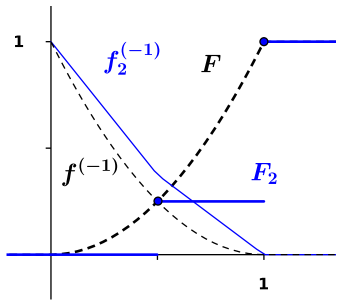

Figure 1: The Williamson 2-transform with illustration of approximation by the convex sum of $m = 2$ Dirac positive distance functions.

where the auxiliary function $g\colon [-\infty, 0] \to [0, 1]$ is defined by $g(x) = f^{(-1)}(-x)$ and $g_-^{(n-1)}$ is the left-derivative of order $n-1$. The inverse transformation is provided by

$$f^{(-1)}(x) = \int_x^\infty \left(1 - \frac{x}{t}\right)^{n-1} dF(t).$$

Approximation and convergence

Let us first illustrate the transform on an example. Consider a positive distance function $F(x) = min(1, x^2)$ and the corresponding density $F'(x) = 2x$ on $[0, 1]$. Then for $x \in [-1, 0]$, $f^{(-1)}(x) = \int_x^\infty \left(1 - \frac{x}{t}\right)^{2-1} dF(t) = \int_x^1 (t-x)\frac{2t}{t} dt = \left[(t-x)^2\right]_x^1 = (1-x)^2$, and $f^{(-1)}(x) = 0$ for $1 < x$. Thus we can write $f^{(-1)}(x) = \max(1-x, 0)^2$ on $[0, \infty]$, and the generator $f(x) = 1 - \sqrt{x}$, $x \in [0, 1]$, is the generator of Clayton copula for parameter $\lambda = -\frac{1}{2}$.

Now consider a function

$$F_2(x) = F\left(\frac{1}{2}\right)\delta_{\frac{1}{2}}(x) + \left(F(1) - F\left(\frac{1}{2}\right)\right)\delta_1(x) = \begin{cases} 0 & x < \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} \le x < 1 \\ 1 & 1 \le x \end{cases}$$

that approximates F by means of a convex sum of $m = 2$ Dirac functions

$$\delta_{x_0}(x) = \begin{cases} 0 & x < x_0 \\ 1 & x \ge x_0 \end{cases}$$

concentrated in respective points $\left(\frac{1}{2}, \frac{1}{4}\right)$, $\left(1, \frac{3}{4}\right)$. Then the Williamson transform with $n = 2$ yields

$$f_2^{(-1)}(x) = \frac{1}{4}\max\left(0, 1 - \frac{x}{\frac{1}{2}}\right) + \frac{3}{4}\max\left(0, 1 - \frac{x}{1}\right) = \begin{cases} 1 - \frac{5}{4}x & x < \frac{1}{2} \\ \frac{3}{4} - \frac{3}{4}x & \frac{1}{2} \le x < 1 \\ 0 & 1 \le x \end{cases}$$

as illustrated on Figure 1.

We can prove that when with $m \to \infty$ the convex sum of Dirac positive distance functions

$$F_m(x) = \sum_{i=1}^{m} p(a_i)\delta_{a_i}(x)$$

(where $p\colon [0,\infty] \to [0,1]$ is the corresponding probability mass function with support in points $0 < a_i$) weakly converges to F then also the related generator for which the pseudo-inverse is given by

$$f_m^{(-1)}(x) = \sum_{x<a_i} p(a_i)\left(1-\frac{x}{a_i}\right)^{n-1} = \sum_{i=1}^{m} p(a_i)\max\left(0, 1-\frac{x}{a_i}\right)^{n-1}$$

converges point-wisely to f and the corresponding Archimedean copula C_m converges point-wisely to C given that $f(x)$ is fixed in, say, $x = \frac{1}{2}$. The fixation is required due to the fact that all multiplications of a generator by some positive constant yield the same copula.

As an example, take the simplest case $a_i = \frac{i}{m}$ and $p(a_i) = \frac{1}{m}$, $i = 1, \dots m$ (evenly spaced and uniformly distributed), we get $f_m^{(-1)}(x) = \sum_{i=1}^{m} \frac{1}{m} \max\left(1 - \frac{mx}{i}\right)$. If $f_m^{(-1)}(x)$ is to converge to $f^{(-1)}(x) = 1 - x + x\log x$ for $x < 1$ and 0 elsewhere, it needs to converge in any point $x \in]0,1[$. Let us examine the convergence e.g. in $x = \frac{1}{2}$,

$$f_m^{(-1)}\left(\frac{1}{2}\right) = \frac{1}{m}\sum_{i=1}^{m}\max\left(1-\frac{m\frac{1}{2}}{i}\right) = \frac{1}{m}\sum_{i=\lfloor\frac{m}{2}\rfloor+1}^{m}\left(1-\frac{m}{2i}\right) = \frac{1}{m}\sum_{i=1}^{\frac{m}{2}}\frac{i}{i+\frac{m}{2}}.$$

The above sum with m approaching infinity equals approximately 0.153426 which is also the value of $f^{(-1)}(1/2)$.

Acknowledgement

The support of the project VEGA 1/0171/12 is kindly acknowledged.

References

[1] A.J. McNeil, J. Nešlehová, Multivariate Archimedean copulas, d-monotone functions and l_1-norm symmetric distributions, Annals of Statistics 37(5B) (2009) 3059–3097.

[2] R.B. Nelsen, An Introduction to Copulas, Springer, New York, 2006.

Conic and Archimax copulas

Michal Dibala
Department of Mathematics,
Faculty of Civil Engineering,
Slovak University of Technology,
Radlinského 11,
813 68 Bratislava, Slovakia
dibala@math.sk
Lucia Vavríková
Department of Mathematics,
Faculty of Civil Engineering,
Slovak University of Technology,
Radlinského 11,
813 68 Bratislava, Slovakia
vavrikova@math.sk

Abstract

The aim of this article is to illustrate connections between the class of conic copulas and the class of Archimax copulas based on the lower Frechet-Hoeffding bound, copula W. For the chosen example we will show the transition from selected Pickands dependence function in definition of Archimax copula to related conic copula. Afterwards we demonstrate the transition from arbitrary conic copula to appropriate Pickands dependence function.

Keywords: Aggregation function, Archimax Copula, Conic Copula, Dependence function.

1 Introduction and preliminaries

In many areas of practice we are encountered with the objective to model relationships between random variables. The copulas theory is one of possible approaches to solve this problem. The tasks with similar philosophy have shown the similar dependence structure, hence many types of copula classes describing specific situations have been formed. At the present time, there exist an inexhaustible quantity of copula classes, and so it is problematic also for experts in mathematical field to choose the correct one. Sometimes it is possible to express the same copula in notion of different copula classes. Therefore it is desirable to know the possibilities of transition between different types of characteristics in terms of various copula classes. Our field of study are copulas for which are characteristics of Archimax copula and conic copula class fulfilled.

This paper is organized as follows. In the second section we introduced basic definitions. In the third section we introduce the transition from Archimax copula class to conic copula class. The fourth section describes reverse transition between copula classes. The last section is the conclusion.

2 Basic definitions

The aggregation problem consists in aggregating n-tuples of objects all belonging to a given set, into a single object of the same set.

First we recall minimal conditions of aggregation operators $A : \bigcup_{n \in N} [0,1]^n \to [0,1]$:

- identity $A(x) = x, \forall x \in [0,1]$,
- monotonicity $A(x_1, \ldots, x_n) \leq A(y_1, \ldots, y_n)$ iff $x_i \leq y_i$, $\forall i \in \{1, \ldots, n\}$, $n \in N$,
- boundary conditions $A(0, \ldots, 0) = 0$ and $A(1, \ldots, 1) = 1$.

The special class of binary aggregation functions with neutral element 1 is the class of copulas.

In this section we recall some basic definitions.

Let $I = [0,1]$, $\mathbf{1} = (1, \ldots, 1)$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

Definition 2.1. *Let $F : \mathbb{R}^2 \to \mathbb{R}$. Then the differences of the function F are:*

$$\triangle_{x_1}^{x_2} F(x,y) = F(x_2, y) - F(x_1, y) \tag{1}$$

$$\triangle_{y_1}^{y_2} F(x,y) = F(x, y_2) - F(x, y_1) \tag{2}$$

Definition 2.2. *Function $F : \mathbb{R}^2 \to \mathbb{R}$ is called 2-increasing if $\forall x_1, x_2, y_1, y_2 \in D(F)$ and $x_1 \leqslant x_2, y_1 \leqslant y_2$ fulfils condition:*

$$\triangle_{x_1}^{x_2} \triangle_{y_1}^{y_2} F(x,y) \geqslant 0. \tag{3}$$

Definition 2.3. *Copula is a 2-increasing function $C : I^2 \to I$ fulfilling conditions:*

- $C(0, v) = C(u, 0) = 0$,
- $C(1, v) = v$,
- $C(u, 1) = u$.

Theorem 2.1. *For all copulas holds:*

$$\underbrace{\max(u+v-1,0)}_{W(u,v)} \leqslant C(u,v) \leqslant \underbrace{\min(u,v)}_{M(u,v)}, \tag{4}$$

where $W(u,v)$ is lower Frechet-Hoeffding limit and $M(u,v)$ is upper Frechet-Hoeffding limit.

Definition 2.4. *Let $\phi : I \to [0,\infty]$ be a continuous strictly decreasing function fulfilling condition $\phi(1) = 0$. Then the function $\phi^{(-1)} : [0,\infty] \to I$, $\phi^{(-1)} = \min(t, \phi(0))$ is pseudo-inverse of ϕ.*

Theorem 2.2. *Let ϕ fulfil condition in Definition 2.4. Then the function $C_\phi : I^2 \to I$, $C_\phi(x,y) = \phi^{(-1)}(\phi(x), \phi(y))$ has the property from Definition 2.3.*

Theorem 2.3. *Let a function f be a additive generator and $A : I^2 \to I$ a convex function with lower limit $\max(x, 1-x)$. Then*

$$C_{f,A}(x,y) = f^{-1}\left(\min(f(0), (f(x)+f(y))\, A\left(\frac{f(x)}{f(x)+f(y)}\right)\right) \tag{5}$$

is a copula.

Two-dimensional Archimax copulas were introduced by Capéraá, Fougéres and Genest [2] as a common extension of both extreme-value copulas and Archimedean copulas.

Definition 2.5. *A set $Z \subset I^2$, $Z_\star \subseteq Z$ is called the zero set, if it is continuous, closed and if $\forall x \in Z$, $u \in I^2$ holds $u \leq x$ than $u \in Z$, where $Z_\star = \{(x,y)|0 \in \{x,y\}\}$.*

The zero set Z_A of an aggregation function A is the inverse image of the value 0, i.e.

$$Z_A := A^{-1}(\{0\}) = \{\mathbf{x} \in [0,1] | A(\mathbf{x}) = 0\}. \tag{6}$$

Since $A(1,\ldots,1) = 1$, Z_A is a proper subset of $[0,1]^n$. A point $\mathbf{x} = (x_1,\ldots,x_n) \in Z_A$ is called weakly undominated point if there exists no $\mathbf{y} = (y_1,\ldots,y_n) \in Z_A$ such that $y_1 > x_1, y_2 > x_2, \ldots, y_n > x_n$. In the case $n = 2$ we will refer to the set of weakly undominated points of the zero-set of a continuous aggregation function as the upper boundary curve of the zero-set [3].

Now follows the general definition of a conic function. We denote the (linear) segment with endpoints $\mathbf{x}, \mathbf{y} \in [0,1]^n$ as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \{\lambda \mathbf{x} + (1-\lambda)\mathbf{y} | \lambda \in [0,1]\}.$$

Definition 2.6. *Let $Z \subset [0,1]^n$ be a closed lower set containing $Z_\star$. We define the function $A_Z : [0,1]^n \to [0,1]$ as follows:*

- $A_Z(\mathbf{1}) = 1$,
- $A_Z(\mathbf{x}) = 0$ *for any* $\mathbf{x} \in Z$,
- *for any weakly undominated point* $\mathbf{x} \in Z$*, the function* A_Z *is linear on the segment* $\langle \mathbf{x}, \mathbf{1} \rangle$.

The function A_Z *is called a conic function with a zero-set* Z.

Theorem 2.4. *[3] Let* Z *be a closed lower set of* $[0,1]^2$ *such that* $Z_\star \subset Z \subseteq Z^\star$ *with corresponding function* $f : [0,d] \to I$, $d \in I$. *The conic aggregation function* A_Z *is a copula if and only if the function* f *satisfies the following conditions:*

- $f(d) = 0$,
- f *is convex.*

The field of our interest are Archimax copulas based on W copula. Additive generator of W is a function $f(x) = 1 - x$, $x \in I$ with defined pseudo-inverse $f^{(-1)}(y) = \max(0, 1 - y)$. From equation (5) we get

$$C_{W,A}(x,y) = \max\left(0, 1 - (2 - x - y) \cdot A\left(\frac{1-x}{2-x-y}\right)\right). \tag{7}$$

It's evident that for $A(z) = 1$, $z \in I$ the formula is

$$C_{W,A}(x,y) = \max\left(0, 1 - (2 - x - y) \cdot 1\right) = \max(0, x + y - 1) = W(x,y).$$

On the contrary when we take

$$A(z) = \begin{cases} 1 - z & \text{if } z \in \left(0, \frac{1}{2}\right], \\ z & \text{if } z \in \left(\frac{1}{2}, 1\right], \end{cases}$$

then we get $C_{W,A}(x,y) = \min(x,y) = M(x,y)$.

For both limit cases the consequent copula falls into the class of conic copulas. In the first case the corresponding function is $f(x) = 1 - x$, $x \in I$, in the second case is $Z_M = Z_x$, where x is weakly undominated point of Z_C. In both cases is consequent copula linear on the segments $\langle x, 1 \rangle$, $x \in I$. Therefore there is a natural question if the choice of any arbitrary convex function $A(z) : I \to I$, $\max(1 - z, z) \le A(z) \le 1$, $\forall z \in I$ we get a conic copula.

3 Transition from the class of Archimax copulas to the class of conic copulas

In this section we will try to verify if in the case of choice of dependence function A the corresponding Archimax copula also belongs to the class of conic copulas.

Let $A(z) = z^2 - z + 1,\ z \in I$ be a convex function and $\max(1-z, z) \leq A(z) \leq 1$. From (7) we get

$$C_{W,A}(x,y) = \max\left(0, 1 - (2-x-y) \cdot \left(\left(\frac{1-x}{2-x-y}\right)^2 - \frac{1-x}{2-x-y} + 1\right)\right) =$$

$$\max\left(0, -\frac{x^2+y^2+xy-2y-2x+1}{2-x-y}\right). \tag{8}$$

Let

$$\frac{x^2+y^2+xy-2y-2x+1}{2-x-y} = 0.$$

We need to obtain the formula of the margin of the zero set from the implicit expression above, therefore $y^2 + (x-2)y + x^2 - 2x + 1 = 0$.

This equation has two solutions:

$$f_1 = 1 - \frac{x}{2} + \sqrt{x - \frac{3}{4}x^2},$$

$$f_2 = 1 - \frac{x}{2} - \sqrt{x - \frac{3}{4}x^2}.$$

It holds $f_1(0) = 1,\ f_1(1) = 0$, therefore f_1 can not describe the zero set $Z_{W_{C,A}}$. For the function f_2 and its convexity holds:

$$\frac{d^2 f_2(x)}{dx^2} = \frac{-\frac{1}{3}}{\left(x - \frac{4}{3}\right) x \sqrt{x - \frac{3}{4}x^2}} > 0, \forall x \in (0,1).$$

So the function f_2 describe the zero set $Z_{W_{C,A}}$.

Now we apply parametrization $y = ax + b$. In that the copula has to be linear on $\langle x, 1\rangle$, we have $ax + b = 1,\ x = 1$ and from this we have $b = 1 - a$. From (8) we prove linearity of the copula

$$\max\left(0, \frac{(a^2+a+1)x^2 - (2a^2+a+1)x + a^2}{(x-1)(a+1)}\right) = \max\left(0, \frac{x - \frac{a^2}{a^2+a+1}}{a+1}\right).$$

Then the roots are:

$$x_1 = 1, x_2 = \frac{a^2}{a^2+a+1}.$$

The consequent copula is linear on $\langle x, 1\rangle$ out of its zero set. For chosen function A we constructed the Archimax copula that is conical simultaneously.

4 Transition from the class of conic copulas to the class of Archimax copulas

Let a conic copula defined as follows [4]:

$$C_f(x,y) = \begin{cases} 0 & 0 \leq x \leq \frac{1}{2}, y \leq (1-2x)^2, \\ \frac{4x(1-x)-1+y}{4(1-x)-1+y} & (1-2x)^2 < y \wedge y \geq 2x-1, \\ \min(x,y) & else. \end{cases}$$

We want to derive a set where the function A attains the constant value.
Let: $z = \frac{1-x}{2-x-y}$, $y \neq 2-x$.

Then holds:

$$y = \frac{1-z}{z}x + \frac{2z-1}{z}, z \neq 0 \tag{9}$$

for chosen constant value z.

From (8) we obtain the function $A(z) = \frac{1-C_f(x,y)}{2-x-y}$.

From (9) we get:

$$A(z) = \begin{cases} 1-z & z \in \left[0, \frac{1}{3}\right), \\ \frac{4z^2}{5z-1} & z \in \left[\frac{1}{3}, 1\right]. \end{cases}$$

The function $A(z)$ is continuous and its convexity proof is easy to show.

Function $A_1(z) = 1-z, z \in \left[0, \frac{1}{3}\right)$ fulfils definition of convexity and $\frac{dA_1\left(\frac{1}{3}^-\right)}{dz} = -1$.

Function $A_2(z) = \frac{4z^2}{5z-1}$ is on $\left(\frac{1}{3}, 1\right]$ double differentiable,

$\frac{d^2A_2(z)}{dz^2} = \frac{8}{(5z-1)^3} > 0, \forall z > \frac{1}{5}$, so $A_2(z)$ is convex and moreover $\frac{dA_2\left(\frac{1}{3}^+\right)}{dz} = -1$, therefore function $A(z) = A_1 \cap A_2$ is convex and fulfils all conditions of Pickands dependence function in definition of Archimax copula.

5 Conclusion

We have demonstrated both transitions between Archimax and conic copulas. Our next work will be concerned on proof of equivalence between conic copula class and Archimax copula class based on W.

Acknowledgment

This work was supported by grant VEGA 1/0171/12 .

References

[1] Calvo T., Kolesárová A., Komorníková M., Mesiar R.: A review of aggregation operators. Servicio de Publicaciones de la U. A. H., Madrid, (2001).

[2] Capéraá, P., Fougéres, A.-L., and Genest, C.: Bivariate distributions with given extreme value attractor. Journal of Multivariate Analysis 72 (2000), pp. 30 - 49.

[3] Jwaid, T.: Semilinear and semiquadratic conjuctive aggregation function. PhD Thesis, Ghent University, Faculty of Sciences 2013-2014, 231, p. 29.

[4] Jwaid, T., De Baets, B., Kalická J., Mesiar, R.: Conic aggregation functions. Fuzzy Sets and Systems. 2011, 167, p. 3-20.

[5] Mesiar, R., Jágr, V.: d-Dimensional dependence functions and Archimax copulas. Fuzzy Sets and Systems. 2013, 228, p. 78-87.

[6] Nelsen. R. B.: An Introduction to Copulas. NY : Springer, 2006. 269 p. ISBN 978-0.387-28659-4

Dynamic Inclusion of New Event Types in Visual Inspection using Evolving Classifiers

Edwin Lughofer

November 20, 2014

Abstract

In this talk, we are dealing with the automatic inclusion of new event types in visual inspection systems. Within the context of image classification for recognizing "OK" and "not OK" parts, a certain event can be directly associated with a class, as events are usually independent and disjoint from each other. In this sense, we are dealing with the problem of integrating a new class into the image classifier on-the-fly, once specified on-line by an operator. We are using evolving fuzzy classifiers (EFC), which are relying on fuzzy rule bases and are able to adapt their structure and update their parameters in incremental manner. The novel methodological aspects lie 1.) in appropriate structural changes in the EFC whenever a new class appears and 2.) in the estimation of the expected change in classifier accuracy on the older classes seen before, which is based on an analysis of the expected change in the classifier's decision boundaries. The second point is an important aspect for operators, as they are already familiar to work with established classifiers that have some accuracy in classification. The new concepts will be evaluated on a real-world visual inspection scenario, where the main tasks is to classify several event types which may occur on micro-fluidic chips and may lead to the deterioration of their quality. The evaluation will be based on two image streams recorded at the inspection system on-line, containing several event types and representing the real production order.

Keywords: visual inspection, new event types, integration of new classes on-the-fly, evolving (fuzzy) classifiers, expected change in classifier's accuracy

Expanding the set of Residual Generators by Genetic-Fuzzy Systems

Francisco Serdio[a]

[a]*Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Austria*

Abstract

Data-driven fuzzy systems as systems models has been proposed in several works into the context of Fault Detection (FD) [1] [2] [3] [4]. Data-driven fuzzy systems exhibit good approximation capabilities because of being proved as universal approximators [5], as they also offer interpretability by readable rules [6], what appears to be important in order to validate the extracted model.

Along our previous research [3] [4], we built fuzzy systems from various data sources, using *sparse fuzzy systems (SparseFIS)* training as described in [7]. This deterministic training algorithm, could produce low quality models for certain process variables when it gets trapped in local minima. When this happens, this models are discarded to conform the set of residuals generators, so faults appearing in this process variables are potentially missed and not detected by the FD framework. To overcome this limitation, we introduce the usage of Genetic Fuzzy Systems (GFSs) [8] [9], which when creating high quality models where SparseFIS was not able to do so, allow to complete the set of residual generators and to increase the performance of the whole residual-based FD approach.

Keywords: residual-based fault detection; fuzzy systems; genetic fuzzy systems; hybridization; black-box modeling

References

[1] M. Shabanian and M. Montazeri, "A neuro-fuzzy online fault detection and diagnosis algorithm for nonlinear and dynamic systems," *International Journal of Control, Automation, and Systems*, vol. 9, no. 4, pp. 665–670, 2011.

[2] E. Laukonen, K. Passino, V. Krishnaswami, G.-C. Lub, and G. Rizzoni, "Fault detection and isolation for an experimental internal combustion engine via fuzzy identification," *IEEE Trans. on Control Systems Technology*, vol. 3, no. 9, pp. 347–355, 1995.

[3] F. Serdio, E. Lughofer, K. Pichler, T. Buchegger, and H. Efendic, "Fault detection in multi-sensor networks based on multivariate time-series models and orthogonal transformations," *Information Fusion*, vol. 20, pp. 272–291, 2014.

Email address: `francisco.serdio@jku.at` (Francisco Serdio)

[4] ——, “Residual-based fault detection using soft computing techniques for condition monitoring at rolling mills,” *Information Sciences*, vol. 259, pp. 304–330, 2013.

[5] J. Castro and M. Delgado, “Fuzzy systems with defuzzification are universal approximators,” *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics*, vol. 26, no. 1, pp. 149–152, 1996.

[6] E. Lughofer, “On-line assurance of interpretability criteria in evolving fuzzy systems — achievements, new concepts and open issues,” *Information Sciences*, vol. 251, pp. 22–46, 2013.

[7] E. Lughofer and S. Kindermann, “SparseFIS: Data-driven learning of fuzzy systems with sparsity constraints,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 396–411, 2010.

[8] O. Cordón, “A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems,” *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 894–913, 2011.

[9] F. Herrera, “Genetic fuzzy systems: taxonomy, current research trends and prospects,” *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27–46, 2008.

Efficient Sensor Placement in Flow Networks and Sensor Networks

Patrick Traxler and **Mariya Zhariy**
Software Competence Center Hagenberg, Hagenberg, Austria
e-mail: {patrick.traxler,mariya.zhariy}@scch.at

Abstract

We study the problem of sensor placement for maximum structural fault detection and isolation in systems with a graphical structure. In particular, we consider flow networks and sensor networks. We are interested in placing as few sensors as possible. We get efficient approximation algorithms and exact algorithms for computing a smallest sensor set for maximum structural fault detectability and isolation.

1 Introduction

In the framework of structural fault diagnosis of Frisk and Krysander [1; 2], systems of mathematical equations describe technical systems. The systems of equations are treated in a purely structural or combinatorial way. In particular, we mainly study the bipartite variables-in-equations graph associated with a system of equations [1]. We use combinatorial concepts and techniques such as the Dulmage-Mendelsohn decomposition [3] and algorithms for computing maximum matchings to solve fault detection, fault isolation, and sensor placement problems [1].

The benefit of this approach is that we can apply it early in the design phase of a system. No sensor data is necessary. In this work we consider systems of equations which have themselves a graphical structure. In particular, we consider flow networks and sensor networks.

Flow networks and models thereof arise in areas such as electrical engineering, hydraulics, and transportation. They essentially contain two building components: Energy nodes (voltage, pressure) which are connected by flow edges (current, volume flow). See Fig. 1 for an example. The mathematical description of flow networks follows general principles. For example, the amount of flow into a node equals the amount of flow out of it, except for sources or sinks. This conservation constraint or flow preservation was formalized as Kirchhoff's current law in the context of electrical circuits.

Sensor networks and models thereof become increasingly important with the advent of the Internet of Things in which systems, machines, and devices are connected via the Internet. One particular type of sensor network that we consider here has the property that all sensors measure the same quantity or similar quantities, e.g. inside or outside temperature, solar radiation, consumed or produced energy. A common way to model sensor networks is to consider the geographic location of sensors. Nearby sensors have often similar values. This can be used for fault detection. See e.g. [4].

We study the problem of placing as few sensors as possible. That is, the results of our algorithms are smallest or almost smallest sensor sets which allow maximum fault isolation. This is in contrast to enumerating all reasonable (i.e. minimal) sensor sets which achieve maximum fault detection or isolation [1]. We also aim at providing running time and approximation guarantees.

We study special cases which are motivated by different models of flow and sensor networks. Our main result reads as follows: The Minimum Sensor Placement (MSP) problem for instances with a symmetric variables-in-equations graph can be solved efficiently and approximatively. As far as flow and sensor networks are concerned, we can model the MSP problem for variable flow networks and sensor networks in terms of symmetric variables-in-equations graphs and can thus solve the problem efficiently. We formulate our special cases in terms of variables-in-equations graphs.

We provide the necessary definitions in Sec. 2, models of flow and sensor networks in Sec. 3, and algorithms and their analysis in Sec. 4.

1.1 Motivating Applications

Flow networks arise naturally in the study of technical systems. Here, we present a snippet of a real-world hydraulic system and show how to apply our results to it. In contrast to many flow networks that describe real-world system, sensor networks may be considerably larger. In our applications, the graphical structure of our flow network has a few dozen of vertices and edges. The snippet of the hydraulic system we present, Fig. 1, has just 5 vertices. The sensor networks can have hundreds of vertices and edges in our application. Sensors in our application measure the produced energy, module temperature, plane-of-array irradiance of photovoltaic systems. But not every photovoltaic system delivers data all the time. In other words, the set of possible sensor locations changes over time. Once per day we check for missing data and place the sensors appropriately.

2 Definitions: Detection, Isolation, Sensor Placement

The purpose of this section is to introduce the Minimum Sensor Placement (MSP) problem.

An example of a set of equations is Eq. 1 and Eq. 2. It describes a part of a hydraulic system. In Fig. 2 we see its variables-in-equations graph. In general, the *variables-in-equations graph* is a bipartite graph $B = (U, V, E)$ with $U \cap V = \{\}$ and U are the equations and V the variables in M. Mathematically, the *system of equations* or *model* M is a set and thus $U = M$. We draw an edge between $e \in U$ and $v \in V$ iff variable v occurs in equation e. Note that $F = \{f_1, \dots, f_6\}$ are the fault variables in the example. Fault variables do not belong to V. Moreover, every fault f occurs in at most one equation which we denote by e_f, and at most one fault is associated with an equation. In the example, we can set $P = \{v_1, \dots, v_6\}$. In general, $P \subseteq V$ is the set of *possible sensor places* or *possible sensor locations* and F is the set of *faults* for which $e_f \in U$ for every $f \in F$.

Dulmage-Mendelsohn Decomposition

For every bipartite graph B there exists an integer n such that the vertices of B and thus M can be partitioned into an under-determined part M_0, a just-determined part M^0, and an over-determined part M^+. The concrete definitions of these parts can be found in [5]. We provide them in Sec. 4. The partition is called the *Dulmage-Mendelsohn (DM) decomposition*[1] and was introduced in [3].

An example of a DM-decomposition with one just-determined part, no under-determined, and no over-determined part is depicted in Fig. 3.

Definition 1 ([1]). *A fault f is structurally detectable in M if $e_f \in M^+$.*

This definition says that we need more equations than variables to detect faults. So, to detect faults for the example in Fig. 3 we need to add one more equation to M. The idea of sensor placement is to add equations of the form $v = c$ for some variable $v \in P \subseteq V$ and a value c. For a set $S \subseteq P$, define M_S as the set of these equations. The goal is to find a small set S such that $e_f \in (M \cup M_S)^+$.

Definition 2 ([1]). *A fault f_i is structurally isolable from fault f_j in M if $e_{f_i} \in (M \setminus \{e_{f_j}\})^+$.*

This notion ensures that we can also identify a fault, i.e. we can locate the fault. Here, we make the assumption that only a single fault happens, i.e. only one fault $f \in F$ can have a value unequal to 0. See [6] for further discussion on this topic and in particular on residual generation. A second assumption is that sensors work correctly. See [6] for a solution to handle faulty sensors.

We are now able to define MSP. As above we want to find a small set S such that $e_{f_i} \in ((M \setminus e_{f_j}) \cup M_S)^+$. Additionally we want to maximize such pairs (i, j). This motivates the following definitions.

Definition 3. *Let $I = (M, F, P)$, $f \in F$, and $S \subseteq P$. Define*

$$\tau_I(f, S) := \{f' \in F : e_{f'} \in ((M \setminus \{e_f\}) \cup M_S)^+\}$$

and

$$\tau_I(f) := \max_{S \subseteq P} \tau_I(f, S).$$

We call a set $S \subseteq P$ for which $\tau_I(f, S) = \tau_I(f)$ for all $f \in F$ a correct sensor placement.

[1] In this work we will mainly cite [5] as it contains a presentation of the DM decomposition which is suitable for our needs. We refer the interested reader to [5] for references of the original work of Dulmage, Mendelsohn and others.

In words, a correct sensor placement is such that it achieves maximum isolation for every fault. It holds that a sensor set $S \subseteq P$ is a correct sensor placement iff S maximizes $\sum_{f \in F} \tau_I(f, S)$. In words, S is a correct sensor placement iff S maximizes the total number of fault pairs (f', f) where f' is structurally isolable from f.

Minimum Sensor Placement (MSP)
Input: Equations M with faults F and sensor locations P.
Output: A smallest set $S \subseteq P$ which is a correct sensor placement.

3 Models for Flow and Sensor Networks

3.1 Flow Networks

Our models of flow networks, see for example Fig. 1, contain energy variables (e.g. pressure, voltage) and flow variables (e.g. volume flow, current.) Energy variables are vertex labels and flow variables are edge labels of a graph G. See Eq. 1 and 2 for an example. One type of equations express flow preservation: The flow into a vertex equals the flow out of the vertex. This is known as Kirchhoff's current law in the context of electrical circuits. An equivalent for hydraulic systems are called pressure built-up equations.

The following example is the snippet of a hydraulic system. It is graphically depicted in Fig. 1. The energy variables p represent pressure. The flow variables c represent the volume flow through some hydraulic component, e.g. pipe, valve, etc.

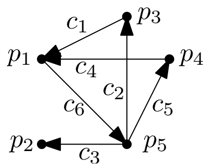

Figure 1: Flow network G_{FN}

The direction of each edge in the flow network graph G_{FN} determines the sign of the corresponding flow summand. The mathematical description of the flow network in Fig. 1 is given by

$$\begin{aligned} e_1 &: & \dot{p}_1 &= c_1 + c_4 - c_6 \\ e_2 &: & \dot{p}_2 &= c_3 \\ e_3 &: & \dot{p}_3 &= -c_1 + c_2 \\ e_4 &: & \dot{p}_4 &= -c_4 + c_5 \\ e_5 &: & \dot{p}_5 &= -c_2 - c_3 - c_5 + c_6 \end{aligned} \tag{1}$$

We assume that the possible sensors are located at the energy variables p_i and that only the flow variables c_j are affected by faults. The variables $f_1, \dots, f_6$ are faults.

$$\begin{aligned} e_6 &: & c_1 &= c_1'(p_1, p_3) + f_1 \\ e_7 &: & c_2 &= c_2'(p_3, p_5) + f_2 \\ e_8 &: & c_3 &= c_3'(p_2, p_5) + f_3 \\ e_9 &: & c_4 &= c_4'(p_1, p_4) + f_4 \\ e_{10} &: & c_5 &= c_5'(p_4, p_5) + f_5 \\ e_{11} &: & c_6 &= c_6'(p_1, p_5) + f_6 \end{aligned} \tag{2}$$

The flow variables c_j depend on the energy variables p_i which are adjacent to c_j in the network flow graph G. The functions $c'(\cdot)$ may be known. But we do not need to know them for sensor placement. It makes however a difference if the functions c' are independent of the energy variables.

The variables-in-equations graph $B = (U, V, E)$ for our example is depicted in Fig. 2: $U = \{e_1, \ldots, e_{11}\}$, $V = \{p_1, \ldots, p_5, c_1, \ldots, c_6\}$ and we draw an edge between $e \in U$ and $v \in V$ if v occurs in e.

We make an interesting case distinction in Fig. 2. On the left, we see the variables-in-equations graph in case of constant flow, i.e. the functions c' are independent of the energy variables. On the right, we see the variables-in-equations graph for variable flow. We call such a bipartite graph symmetric which has the same structure if we exchange the left and right side. We provide the exact definition in Sec. 4.3.

We can easily generalize the above sample derivation of the set of equations from our graph in Fig. 1 to arbitrary graphs G. We call these systems *constant flow networks* if the functions $c'(\cdot)$ are constants and *variable flow networks* in the other case.

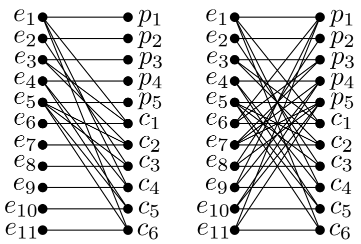

Figure 2: Variables-in-equations graphs for the flow network examples with constant flow (left) and variable flow (right), resp.

3.2 Sensor Networks

In sensor networks, we model variables as the vertex labels of an undirected graph G. Equations are of the form $x_i = f(X_i)$ where X_i is the set of adjacent variables to x_i in G. See Fig. 3 for an example.

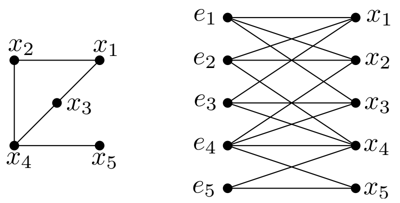

Figure 3: Sensor network G_{SN} (left) and its variables-in-equations graph $B = B(M_{\text{SN}})$ (right)

We can derive the system of equations directly from the graph G in Fig. 3. We have 5 equations in our example. A sample equation is e_1 with $X_1 = \{x_2, x_3\}$: $x_1 = g_1(x_2, x_3)$ if $f_1 \notin F$ and $x_1 = g_1(x_2, x_3) + f_1$ if $f_1 \in F$. We may read the latter equation in two ways: The derivation of the value of x_1 from its neighboring variables in G is possibly faulty. Or, the value of x_1 is possibly faulty.

In applications, the measured quantities are for example temperature, solar radiation, consumed or produced energy. The functions g_i are often linear in the variables from X_i.

In the above way we can derive for a given sensor network G_{SN} with variables $x_1, \ldots, x_n$ and faults F a model of sensor networks suitable for fault detection and isolation. We call such a system a *sensor network*.

4 Algorithms: Description and Analysis

4.1 Sensor Placement as Graph Reachability

We show how sensor placement reduces to graph reachability. This is similar to Lemma 1 in [1]. We also provide the formal definition of the DM decomposition here. The presentation follows [5].

We recall that $X \subseteq E$ is a *(perfect) matching* in a bipartite graph $B = (U, V, E)$ if every vertex in B occurs in (exactly one) at most one edge in M. Let X be a maximum matching in B. An *alternating path* w.r.t. X is a path in B such that no two neighboring edges in it are both from X or both from $E \setminus X$. A vertex is called *matched* by X if it occurs in X and *unmatched* otherwise. We define VR (HR) as the set of all vertices from U which are reachable from some unmatched vertex in U (V) via some alternating path w.r.t. X. We define VC (HC) as the set of all vertices from V which are reachable from some unmatched vertex in U (V) via some alternating path w.r.t. X. Set SR := $U \setminus (\text{HR} \cup \text{VR})$ and SC := $V \setminus (\text{HC} \cup \text{VC})$.

Let M be a set of equations and $B = B(M) = (U, V, E)$ the corresponding variables-in-equations graph. Since $|\text{HR}| < |\text{HC}|$, $|\text{SR}| = |\text{SC}|$, and $|\text{VR}| > |\text{VC}|$ (see [5]), we call the respective subgraphs of B which are induced by HR $\cup$ HC, SR $\cup$ SC, and VR $\cup$ VC as the under-determined part M_0, the just-determined part M^0, and the over-determined part M^+. This partition of B and thus M is independent of the choice of X, see Theorem 2.1 in [5].

Proposition 1. *Any two maximum matchings X_1 and X_2 in a bipartite graph B yield the same sets* HR, HC, VR, VC, SR, SC.

This completes the formal definition of structural detectability, Def. 1, and structural isolability, Def. 2.

Next, we define for B with a perfect matching X the graph $G(B, X)$. It is B where edges not in X are directed from equation vertices U to variable vertices V and edges in X are shrunk into a single vertex. Thus, $G(B, X)$ is a directed graph and we can naturally identify the vertices in $G(B, X)$ from U and V.

Let us consider some example. We derive the variables-in-equations graph $B_{\text{SN}} = B(M_{\text{SN}})$ in Fig. 3 from the sensor network G_{SN}. A maximum and actually a perfect matching X in B is given by the edges $\{e_i, x_i\}$, $i \in \{1, \ldots, 5\}$. We make two observations.

First, $G(B_{\text{SN}}, X)$ is equivalent to G_{SN} in our example. The difference is that $G(B_{\text{SN}}, X)$ does have edge orientations. Second, there is no under-determined part M_0 and no over-determined part M^+ in B_{SN}. In particular, SR = SC = {} and thus no fault is detectable.

Adding sensor measurement equations M_S, $S \subseteq P \subseteq V$, to M_{SN} is equivalent to adding new vertices to U in B_{SN}. In our example, the measurement of a single sensor variable suffices to yield maximum fault detectability. We just note that the perfect matching in our example is a maximum matching in $M \cup M_S$.

This result holds in general if we have a set of equations M and $B = B(M)$ has a perfect matching. The situation is however more complicated if we first remove an equation from M and then add equations for a sensor measurement – the situation that arises in case of fault isolation.

We will need the concept of an augmenting path. We recall that an *augmenting path* p of a matching X is an alternating path w.r.t. X that begins and ends with vertices which are unmatched by X (see e.g. [5].) We observe that the symmetric difference $p \oplus X$ is a new matching of size $|X| + 1$.

Lemma 1. *Let M be a set of equations and $B = B(M) = (U, V, E)$ its variables-in-equations graph. Let F be the set of faults and P be a set of possible sensor locations. Assume that B is connected and has a perfect matching X. For every fault $f \in F$, let v_f be the corresponding vertex in $G(B, X)$.*

1. Let $f \in F$ and $S \subseteq P \subseteq V$. The fault f is structurally detectable in $M \cup M_S$ iff there exists $s \in S$ such that v_f is reachable from s in $G(B, X)$.

2.a. Let $f, f' \in F$ and $S \subseteq P \subseteq V$. Assume that the size of a maximum matching in $B(M \setminus \{e_f\} \cup M_S)$ is $|X| - 1$. The fault f' is structurally isolable from f in $M \cup M_S$ iff there exists $s \in S$ such that $v_{f'}$ is reachable from s in $G(B(M), X) \setminus \{v_f\}$.

2.b. Let $f, f' \in F$ and $S \subseteq P \subseteq V$. Assume that the size of a maximum matching in $B(M \setminus \{e_f\} \cup M_S)$ is $|X|$. The fault f' is structurally isolable from f in $M \cup M_S$ iff there exist $s, t \in S$ such that $v_{f'}$ is reachable from s in $G(B_{f,t}, X_t)$ with $B_{f,t} := B(M \setminus \{e_f\} \cup M_{\{t\}})$. Here, X_t is a perfect matching in $B_{f,t}$ and it emerges from $X' := X \setminus \{e_f\}$ via an augmented path p of X' in $B_{f,t}$, i.e. $X_t = X' \oplus p$.

Proof. We start with (1). Let e_f be the equation associated with f. By the definition of structural fault detection, fault f is detectable if $e_f \in M^+$ with $M := M \cup M_S$. We observe that X is a maximum matching in $B(M \cup M_S)$, i.e. the addition of a measurement equation will not increase the size of a matching. Moreover, there is a one-to-one correspondence between an alternating path in $B(M)$ w.r.t. X and a path in $G(B(M), X)$.

We continue with (2). By the definition of structural fault isolation, fault f' is isolable from fault f if $e_{f'} \in M'^+$ with $M' := (M \setminus \{e_f\}) \cup M_S$. Let X' be X with the edge removed which contains e_f. We observe that X' is a maximum matching in $B(M \setminus \{e_f\})$ and that a maximum matching in $B(M')$ either has size $|X'|$ or $|X'| + 1$.

Assume the former, case (2.a). This is the simple case since X' is a maximum matching in $B(M')$. Thus, reachability in $B(M')$ via an alternating path w.r.t. X' starting from unmatched equations in $B(M')$ corresponds to reachability in $G(B(M), X') \setminus \{v_f\}$ starting from a vertex in S. Just note that the unmatched equations in $B(M')$ are the measurement equations M_S and that every measurement equation vertex is connected to exactly one variable vertex.

Assume that the size of a maximum matching in $B(M')$ is $|X'| + 1$, case (2.b). This case is more complicated since a maximum matching in $B(M')$ can be rather different from X'. However, the following holds: There exists an augmenting path p of X' such that the induced matching $X'' = p \oplus X'$ is maximum in $B(M')$. This result is known as Berge's Lemma. We observe that p starts with an unmatched equation vertex $e \in M_S$. It is connected to a variable vertex. Let t be this vertex. Thus, $X_t = X''$ is maximum in $B(M')$ and a perfect matching in $B_{f,t}$. The claim follows from the above correspondence between alternating paths in $B_{f,t}$ w.r.t. X_t and paths in $G(B_{f,t}, X_t)$. □

4.2 Sensor Placements as Hitting Sets

We show how sensor placement reduces to the computation of hitting sets. It is similar to Theorem 2 in [1]. In particular, we are going to show that a sensor set $S \subseteq P$ is a correct sensor placement, Def. 3, iff S is the hitting set of a set system $\mathcal{D}$, i.e. $S \cap D \neq \{\}$ for all $D \in \mathcal{D}$.

We start with some simplifications. Let $I = (M, F, P)$. In the definition of $\tau_I(f)$ we take the maximum over all $S \subseteq P$. It follows from the definition of fault isolation that, for fixed $f \in F$, $\tau_I(f, \cdot)$ is monotone, i.e., $\tau_I(f, S) \leq \tau_I(f, S')$ if $S \subseteq S'$.

Proposition 2. *It holds that $\tau_I(f) = \tau_I(f, P)$.*

The following lemma characterizes correct sensor placements as hitting sets. We describe the algorithm `Reduce` first. Its input is a model M and a perfect matching X of $B(M)$. It computes $G(B, X)$ from M and X. For every fault $f \in F$, it checks if an augmenting path p of $X \setminus \{e_f\}$ exists in $B(M \setminus \{e_f\} \cup M_S)$. If no augmenting path exists we set $G' := G(B, X) \setminus \{v_f\}$. Otherwise, we set $G' := G(B_{f,t}, X_t)$ with $B_{f,t}$, X_t as in Lemma 1, and t is some vertex from P such that we can reach v_f from t. Define $D_{f,f'}$ as the set of all vertices $s \in P$ in G' such that $v_{f'}$ is reachable from s. The output of `Reduce` is

$$\mathcal{D} := \{D_{f,f'} : f, f' \in F,\ D_{f,f'} \neq \{\}\}.$$

Lemma 2. *Let M be a set of equations and B its variables-in-equations graph. Let F be the set of faults and P be a set of possible sensor locations. Assume that B is connected and has a perfect matching X. A set $S \subseteq P$ is a correct sensor placement iff S is a hitting set of*

$$\mathcal{D} := \{D_{f,f'} : f, f' \in F,\ D_{f,f'} \neq \{\}\}.$$

Moreover, algorithm `Reduce` computes $\mathcal{D}$ in time $O(n^2 m)$ where n is number of variables and m is the number of variable occurrences in M.

Proof. By Proposition 2 and Definition 3, a sensor set $S \subseteq P$ is a correct sensor placement iff $\tau_I(f, S) = \tau_I(f) = \tau_I(f, P)$ for all $f \in F$. For every $f \in F$, $\tau_I(f, P) = |\{f' \in F : D_{f,f'} \neq \{\}\}|$. Here, we apply Lemma 1. In words, a hitting set of $\mathcal{D}$ satisfies the properties of a correct sensor placement and vice versa.

The running time follows since there are $O(n^2)$ pairs $(f, f') \in F \times F$ and since a depth-first search can be done in time $O(m)$. Note that m is the number of edges in B and G. Moreover, we can search for the augmenting paths in advance. Since an augmenting path is an alternating path, the search reduces to a reachability problem, i.e. a depth-first search. We can do this in time $O(nm)$. □

The reduction is almost optimal since $\mathcal{D}$ can contain $\Omega(n^2)$ sets with at least $\Omega(n)$ elements each. A possible improvement would be $O(n^3 + m)$. We also note that our reduction improves upon the slightly more general reduction in [1]. It has a running time of $O(n^{2.5} m)$. See also [7]. We achieve this by searching for an augmenting path instead of computing a maximum matching.

Let $S^* \subseteq P$ be a smallest correct sensor placement of some MSP instance. We call a set $S \subseteq P$ a c-approximate solution if $|S| \leq c \cdot |S^*|$ and S is a correct sensor placement.

Theorem 1. *There is a polynomial time algorithm for MSP that, given an instance (M, F, P) of MSP such that $B(M)$ is connected and has a perfect matching, outputs a $O(\log(n))$-approximate solution.*

Proof. The one-to-one correspondence in Lemma 2 between correct sensor placements and hitting sets allows us to apply any algorithm which computes exact or approximate solutions of minimum hitting sets in the following way. We first compute a perfect matching (see e.g. [8] pg. 664) and then apply `Reduce`. The result is a set system. We then apply the algorithm in [9] for computing an approximate solution to MHS. It has an $O(\log(n))$ approximation guarantee. □

4.3 Symmetric Variables-In-Equations Graphs

In this section we consider the case of symmetric variables-in-equations graphs. Examples are variable flow networks as defined in Sec. 3.1 and sensor networks as defined in Sec. 3.2. In particular, sensor networks motivate the results in this section. We aim at deriving an efficient algorithm which is capable to solve MSP for instances up to some thousands of sensor locations.

We call a bipartite graph $B = (U, V, E)$ *symmetric* w.r.t. to some perfect matching X of B if for all $\{i, j\} \in E$ there exists vertices k and l such that $\{k, l\} \in E$, $\{i, k\} \in X$, and $\{j, l\} \in X$. See Fig. 3 for an example of a symmetric bipartite graph.

A consequence of symmetry is the following: For every edge e in the directed graph $G(B, X)$ as defined in Sec. 4.1, $G(B, X)$ also contains an edge with the opposite orientation. Thus, reachability in the directed graph $G(B, X)$ is equivalent to reachability in the undirected graph $G^u(B, X)$ which emerges from $G(B, X)$ by removing edge orientations.

For our example of a sensor network, Sec. 3.2, we can summarize the graph transformations as follows: We start with an undirected graph G_{SN} which describes the structure of the sensor network. We derive the model M_{SN} and its bipartite variables-in-equations graph $B = B(M_{\text{SN}})$. A perfect matching X is naturally given. We derive $G(B, X)$ and thus $G^u(B, X)$. The graph $G^u(B, X)$ is identical to G_{SN}.

We will need the concept of biconnected components. We recall that a *biconnected component* C of an undirected graph G is a maximal biconnected subgraph B, i.e. removing any vertex in B will yield a connected subgraph of B. If two biconnected components have a vertex v in common we call it a *cut vertex*. Removing a cut vertex in G, yields a graph with at least two connected components. Removing a non-cut vertex in a connected graph, yields a connected graph. We call an undirected graph biconnected iff it contains only one biconnected component.

Lemma 3. *Let M be a set of equations and $B = B(M)$ its variables-in-equations graph. Let F be the set of faults and P be a set of possible sensor locations. Assume that B is symmetric.*

1. If $G^u(B)$ is connected, every sensor $s \in P$ achieves maximum fault detectability[2].

2. If $G^u(B)$ is biconnected, any two sensors $\{s, t\} \subseteq P$ are a correct sensor placement.

Proof. As for case (1), we can directly apply Lemma 1. For case (2), we make the case distinction as in Lemma 1. Let $f, f' \in F$ and $S \subseteq P \subseteq V$ and assume that the size of a maximum matching in $B(M \setminus \{e_f\} \cup M_S)$ is $|X| - 1$; case (2.a). Then, fault f' is structurally isolable from f in $M \cup M_S$ iff there exists $s \in S$ such that $v_{f'}$ is reachable from s in $G(B(M), X) \setminus \{v_f\}$. The latter is true due to 2-connectedness of $G^u(B)$.

For the second case, we have to check if there exist $s, t \in S$ such that $v_{f'}$ is reachable from s in $G(B_{f,t}, X_t)$ with $B_{f,t} := B(M \setminus \{e_f\} \cup M_{\{t\}})$. Here, X_t is a perfect matching in $B_{f,t}$. It emerges from $X' := X \setminus \{e_f\}$ via an augmented path P of X' in $B_{f,t}$, i.e. $X_t = X' \oplus P$. The difficulty is that the graph $G(B_{f,t}, X_t)$ is no longer undirected since $B_{f,t}$ is not symmetric. However, it holds that $G(B_{f,t}, X_t)$ consists of single strongly connected component plus an additional vertex. To see this, we show that the vertices of P lie on a directed cycle in $G(B_{f,t}, X_t)$. First, we recall that P corresponds to some path in $G^u(B)$ that starts with t and ends with v_f. Second, due to the biconnectedness of $G^u(B)$ there exists two vertex-disjoint (undirected) paths P_1, P_2 from v_f to t. This result is known as Menger's Theorem. We set P such that it corresponds to P_1. The difference between $G^u(B)$ and $G(B_{f,t}, X_t)$ are the vertices of P_1. Because of the alternating path P, the bipartite graph $B_{f,t}$ and thus $G(B_{f,t}, X_t)$ changes. There is a correspondence between the vertices in $G(B_{f,t}, X_t)$ which are affected by P and the vertices in $G^u(B)$ which lie on P_1. Moreover, the application of the augmenting path P will change the orientation. In the resulting graph $G(B_{f,t}, X_t)$ there are vertices u and v and a directed path from u to v. The vertices u and v correspond to the end and the vertex after the start vertex of P. We can make this path a cycle by using P_2. This implies that the connectedness is preserved in $G(B_{f,t}, X_t)$. Moreover, all the sensor and fault locations are preserved in $G(B_{f,t}, X_t)$. We conclude for any fault $f' \neq f$, $f' \in F$ is detectable since $v_{f'}$ is reachable from s in $G(B_{f,t}, X_t)$. (See Fig. 4 for an example.) □

In Fig. 4 we describe on an example what happens if we remove one equation $e_2 = e_f$ from a model M_{SN} of a sensor network. The sensor network G_{SN} is depicted on the left in the figure. The natural perfect matching X in our sensor network is $\{e_i, x_i\}$ for $i \in \{1, \dots, 4\}$. After removing the edge from X which contains e_2 we need to find a new maximum matching. In our example in Fig. 4 we start at the sensor measurement equation m_2 and compute an alternating path to x_2. Note that both m_2 and x_2 are unmatched by the natural perfect matching X without e_2. We thus have an augmenting path. We compute the new maximum matching from the augmenting path. It is depicted by bold lines in the figure (middle). On the right we see the resulting graph. The variables-in-equations graph is no longer symmetric in general. In our example, there is only an edge from e_1x_1 to e_4x_3 but not in the other direction. We observe that the digraph consists of a single strongly connected component with the exception of vertex m_2x_4. Also note that all the sensor locations are preserved. Thus, since we have a second sensor measurement equation in our example, we can detect every remaining fault, in particular from e_4x_3.

[2] A sensor set $S \subseteq P$ in a model M achieves maximum detectability iff $(M \cup M_S)^+ = (M \cup M_P)^+$, i.e. we can structurally detect the same set of faults if place them at sensor locations S or if we add all possible sensors.

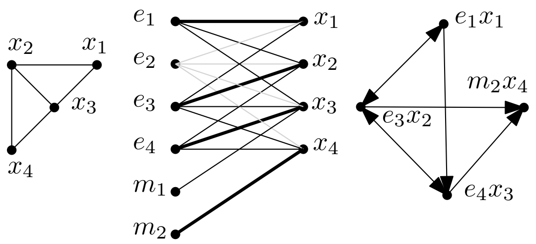

Figure 4: A biconnected sensor network G_{SN} (left) with $P = \{x_3, x_4\}$, its variables-in-equations graph $B_{f,t}$ after removing e_2 and with a new maximum matching X_t (middle), and $G(B_{f,t}, X_t)$ with a directed cycle (right).

The following theorem is a direct consequence of the lemma: If $S \subseteq P$ shares at least two sensors places with every biconnected component in $G^u(B, X)$, then S is a correct sensor placement. We also use the algorithm of Hopcroft and Tarjan [10] for the computation of biconnected components. Its running time yields the claimed running time.

Theorem 2. *There is an algorithm for MSP that, given an instance (M, F, P) of MSP such that $B(M)$ is connected, symmetric and such that every biconnected component $B_1, \ldots, B_k$ of $G^u(B, X)$ has at least two sensor locations, outputs a k-approximate solution. The running time is $O(n + m)$ where n is the number of variables and m is the number of variable occurrences in M.*

4.4 Maximum Detectability of Constant Flow Networks

So far we dealt with placing sensors to achieve maximum fault isolation. In this section we demonstrate how to apply Lemma 2 to study the special case of maximum fault detection for constant flow networks. The algorithm is `Reduce`. In the proof of the theorem we show that `Reduce` outputs a set system which is a graph. We use here that sensors can be placed at any energy variable and that faults can happen at any flow variable. This leads to a particular structure of a constant flow network as depicted in Fig. 2 on the left.

Theorem 3. *There is an algorithm that, given a constant flow network (M, F, P), outputs a sensor set $S \subseteq P$ which achieves maximum fault detection and is at most 2 times larger than the smallest such sensor set. Its running time is $O(n^2 m)$ where n is number of variables and m is the number of variable occurrences in M.*

Proof. Let M_{CFN} be some constant flow network with the perfect matching X which is naturally given by M_{CFN}. The claim of the proposition follows from Lemma 2. We just have to observe that $|D| \leq 2$ for every $D \in \mathcal{D}$. This is due to the partition of the vertices of $G = G(B(M_{\text{CFN}}), X)$ into V_1 and V_2: V_1 are the vertices where sensors can be placed and V_2 are the vertices where a fault can happen. Moreover, every vertex v in V_2 has in-degree exactly two. These two neighbors, v_1 and v_2, are from V_1 and G contains only edges from v_1, v_2 to v. The vertices v_1 and v_2 have in-degree 0 and are thus not reachable from any other vertices in V_1. (See also Fig. 2, left.)

Thus, we have an instance of Minimum Vertex Cover. A 2-approximation algorithm can be found e.g. in [8], pg. 1024. □

5 Conclusion

We provided efficient algorithms for sensor placement in flow and sensor networks in the framework of Frisk and Krysander [1]. We showed how to reduce the study of fault detection and isolation to graph reachability. Our reduction runs in time $O(n^2 m)$. We also used the concepts of graph reachability to design and analyze an efficient approximation algorithm for the case of symmetric variables-in-equations graphs. Our algorithm runs in time $O(n+m)$ and is thus able to handle moderately large instances.

Although our results are tailored towards studying particular models of flow networks and sensor networks, we think that our approach makes it easy to study sensor placement for other special cases of sensor placement too. For example, we presented another application of our approach to study maximum fault detection of constant flow networks.

References

[1] Erik Frisk and Mattias Krysander. Sensor placement for maximum fault isolability. In *18th International Workshop on Principles of Diagnosis, 2007*, pages 106–113, 2007.

[2] Mattias Krysander and Erik Frisk. Some theoretical results on sensor placement for diagnosis based on fault isolability specifications. Technical Report LiTH-R-2770, Linköpings Universitet, 2007.

[3] A. L. Dulmage and N. S. Mendelsohn. Covering of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

[4] Yang Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, 12(2):159–170, 2010.

[5] Alex Pothen and Chin-Ju Fan. Computing the block triangular form of a sparse matrix. *ACM Trans. Math. Softw.*, 16(4):303–324, December 1990.

[6] Mattias Krysander. *Design and Analysis of Diagnosis Systems Using Structural Methods*. Linköpings Universitet, June 2006.

[7] Yi Dong and Gautam Biswas. Comparison for sensor placement algorithms. In *24th International Workshop on Principles of Diagnosis, 2013*, pages 53–58, 2013.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2 edition, 2001.

[9] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[10] John Hopcroft and Robert Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973.

Domain Generalization based on Transfer Component Analysis

Thomas Grubinger[(1)], **Adriana Birlutiu**[(1,2)], **Holger Schöner**[(1)], **Tom Heskes**[(3)]
[(1)]Data Analysis Systems, Software Competence Center Hagenberg,
Softwarepark 21, 4232 Hagenberg, Austria
[(2)]Faculty of Science, "1 Decembrie 1918" University of Alba-Iulia, Romania
[(3)]Institute for Computing and Information Sciences, Radboud University Nijmegen, Netherlands

1 Abstract

In many real-world applications one would like to make use of the knowledge acquired from related domains on previously unseen domains. This problem is known as *domain generalization*, and recently has started to gain attention in the machine learning community [3, 1]. *Domain adaptation* [5] and domain generalization are subareas of transfer learning, aiming to find a shared subspace for related domains. While domain adaptation methods require at least some input data from the target domains, domain generalization methods are designed to generalize to previously unseen domains.

Transfer Component Analysis (TCA) [5] is a domain adaptation technique that aims to learn a shared subspace between a source domain and a target domain. The shared subspace consists of some transfer components learned in a *reproducing kernel Hilbert space (RKHS)* [4] using *maximum mean discrepancy (MMD)* [2]. In the subspace spanned by these transfer components, data distributions of different domains are close to each other and data properties are preserved.

In this paper, we extend the formulation of TCA to multiple domains. Besides the possibility of using this extension in domain adaptation problems we propose to use it for domain generalization as well. Our solution is based on the idea of learning a shared subspace between source domains and using this subspace for related target domains – without re-training. We present and evaluate two variants of our extension, an unsupervised version to which we refer as *Multiple-Domain Transfer Component Analysis (Multi-TCA)* and a semi-supervised version called *Multiple-Domain Semi-Supervised Transfer Component Analysis (Multi-SSTCA)*.

Acknowledgment Parts of this work were carried out in mpcEnergy, a project supported within the program Regionale Wettbewerbsfähigkeit OÖ 2007-2013 by the European Fund for Regional Development as well as the State of Upper Austria.

References

[1] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NIPS*, pages 2178–2186, 2011.

[2] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two-sample-problem. In *NIPS*, pages 513–520, 2006.

[3] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the 30th International Conference on Machine Learning*, pages 10–18, 2013.

[4] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[5] S.J. Pan, I. Tsang, J. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.